Support for Large Datasets in EOSDIS

Mike Folk, NCSA

Professor Marianne Winslett, Dept of Computer Science

University of Illinois at Urbana-Champaign

http://hdf.ncsa.uiuc.edu:8001/nra/nraproject.html
http://bunny.cs.uiuc.edu/CADR/arrays.html
mfolk@ncsa.uiuc.edu
winslett@bunny.cs.uiuc.edu

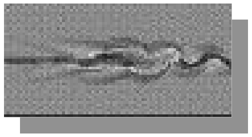
This project has two parts

- **NCSA work**
  - **Next-generation HDF**
  - **Compression and chunking in HDF**
  - **HDF in a Network of Workstations**
- **CS Department work (separate presentation)**
  - **Server-directed collective I/O**

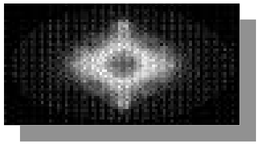
Next Generation: BigHDF

- **New data model**
  - **new primitive number types**
  - **unified data object type**
- **New file structure**
  - **file = boot block + object(s)**
  - **hierarchies of objects**

Current HDF Object Types



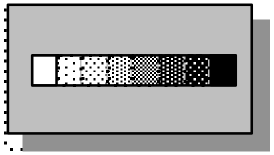
8-bit raster



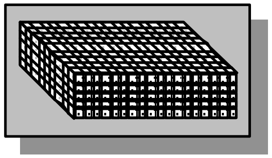
24-bit raster

This HDF file contains one example of each object type

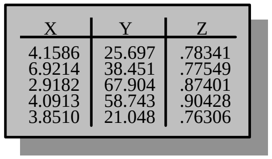
Annotation



Palette

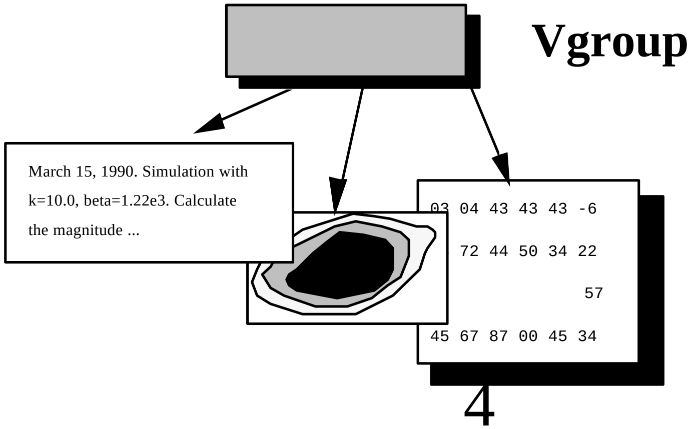


Scientific Data Sets (SDS) (Multi-dimensional arrays)



X	Y	Z
4.1586	25.697	.78341
6.9214	38.451	.77549
2.9182	67.904	.87401
4.0913	58.743	.90428
3.8510	21.048	.76306

Vdata (tables)



March 15, 1990. Simulation with k=10.0, beta=1.22e3. Calculate the magnitude ...

```
03 04 43 43 43 -6

   72 44 50 34 22

                57

45 67 87 00 45 34
```

Vgroup

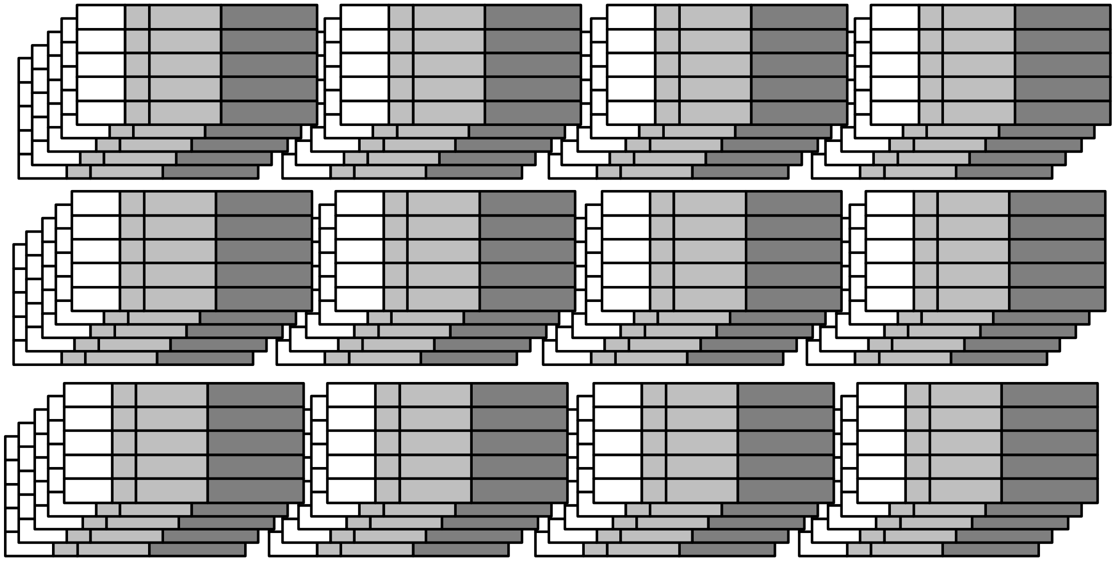
Primitive number types

- **simple**
  - **the usual scalars**
  - **complex, date, time**
- **pointers**
- **structures: any collection of primitives and/or arrays of primitives**

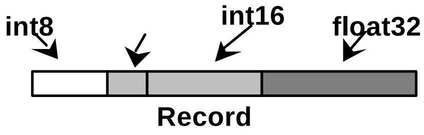
One basic data object type

- **multidimensional array of atomic types**
- **all other objects (raster, SDS, etc.) are a subtype of this type**
- **User-defined attribute list**
- **Special storage options: external, extendable, chunked, sparse (indexed?), compressed**

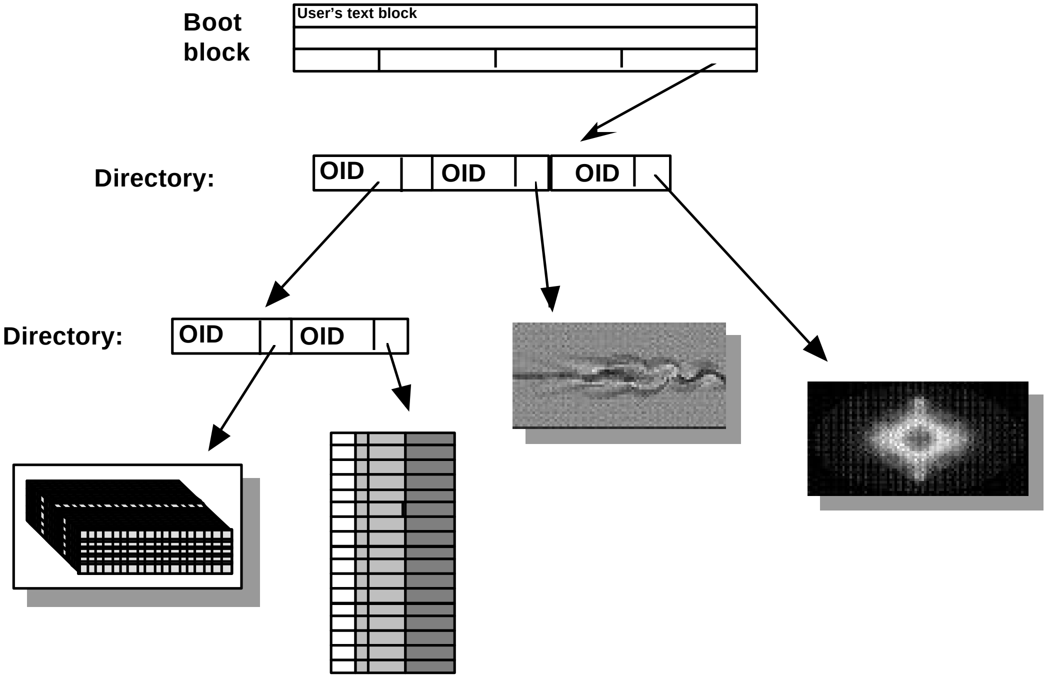
Multidimensional array of atoms



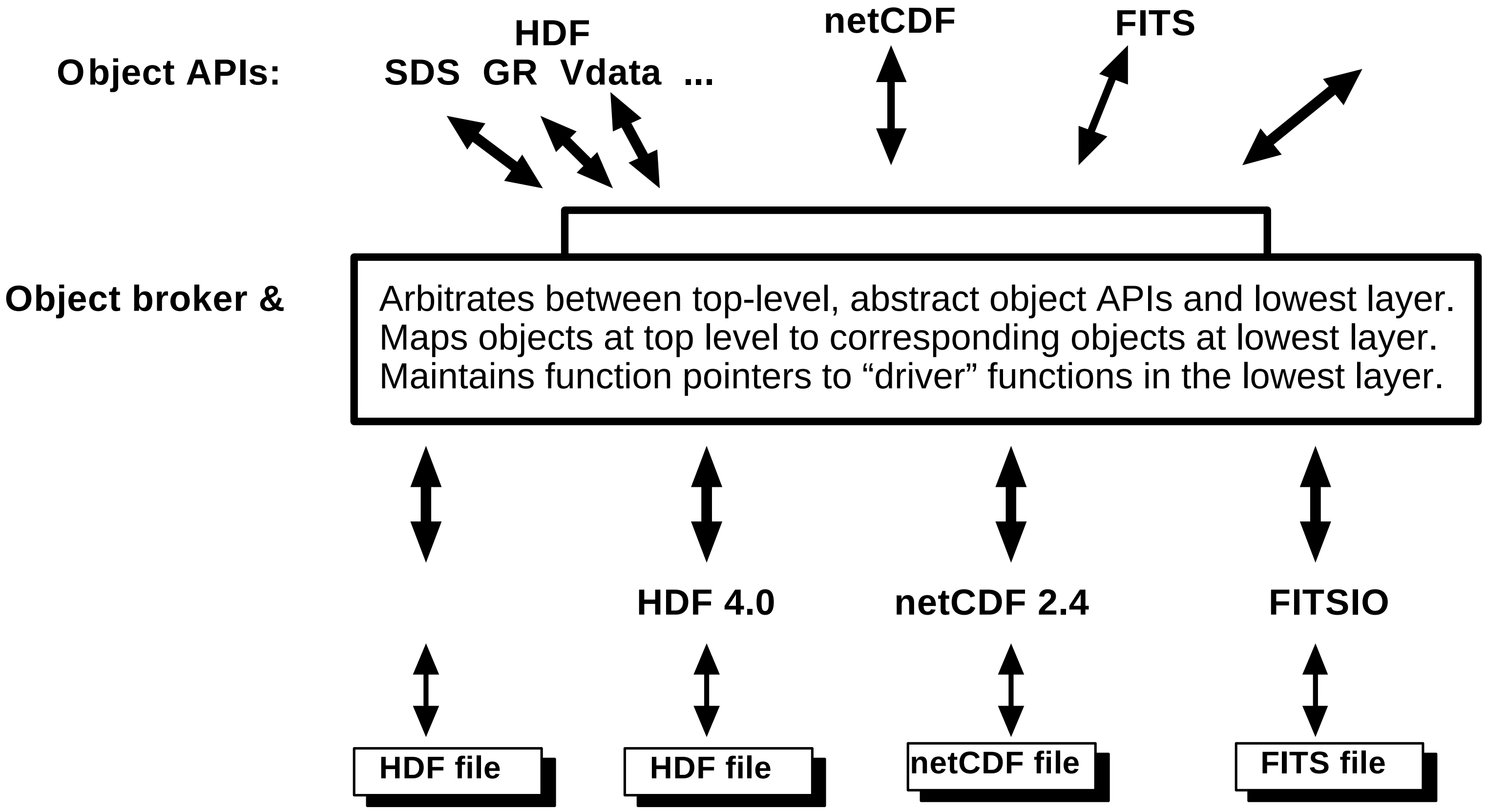
Dimensionality: 5 x 3 x 4



File structure



Library implementation



Object APIs:

HDF
SDS GR Vdata ...

netCDF

FITS

Object broker &

Arbitrates between top-level, abstract object APIs and lowest layer.
Maps objects at top level to corresponding objects at lowest layer.
Maintains function pointers to “driver” functions in the lowest layer.

HDF 4.0

netCDF 2.4

FITSIO

HDF file

HDF file

netCDF file

FITS file

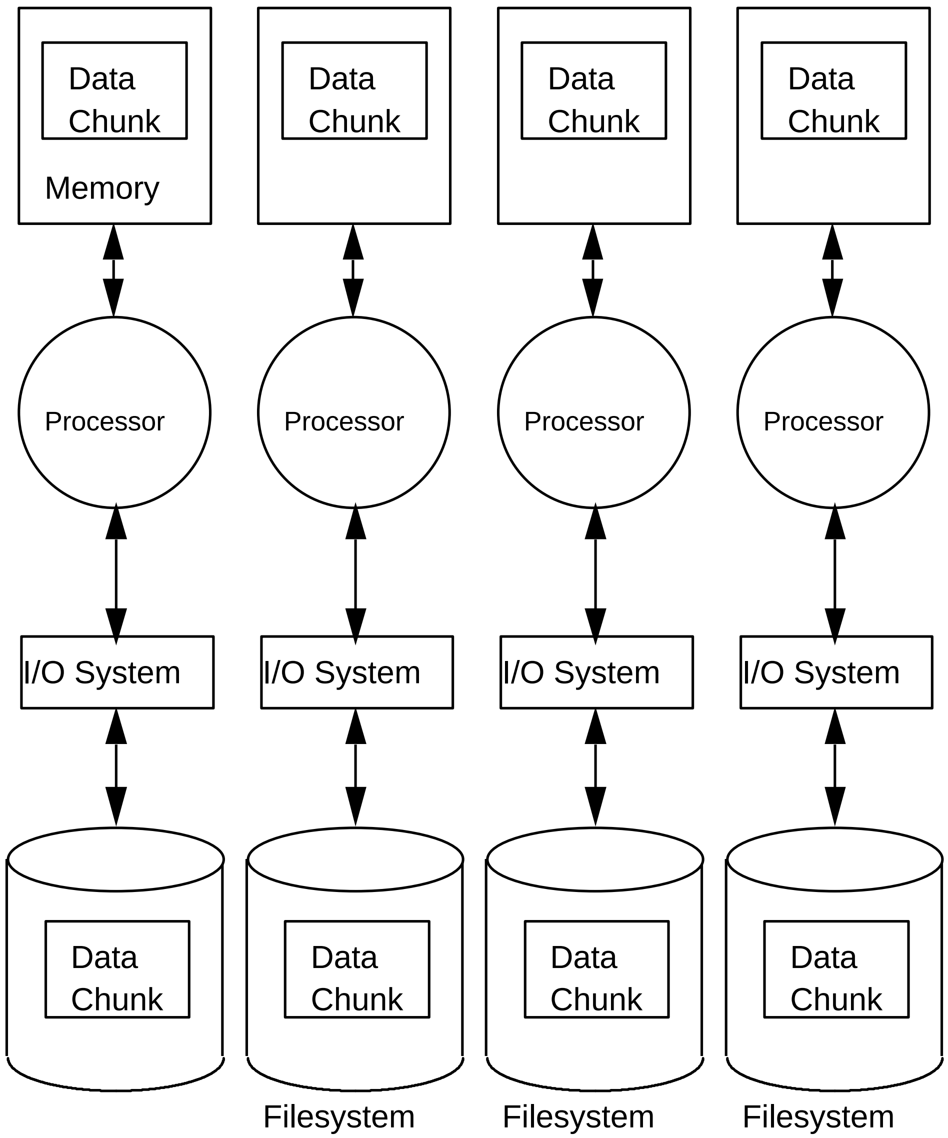
Compression

- Add support for data compression in HDF
- Combine data compression with chunking
- Modular approach: add schemes as needed
- Work completed:
  - Adaptive Huffman & GZIP compression
  - Infrastructure for chunking

HDF and NOW Access

- **Investigate ways to use HDF in Network of Workstations (NOW) environment**
- **Prototype implementation of HDF in NOW environment**

Architecture of NOW



Data Chunk

Memory

Data Chunk

Data Chunk

Data Chunk

Processor

Processor

Processor

Processor

I/O System

I/O System

I/O System

I/O System

Data Chunk

Data Chunk

Data Chunk

Data Chunk

Filesystem

Filesystem

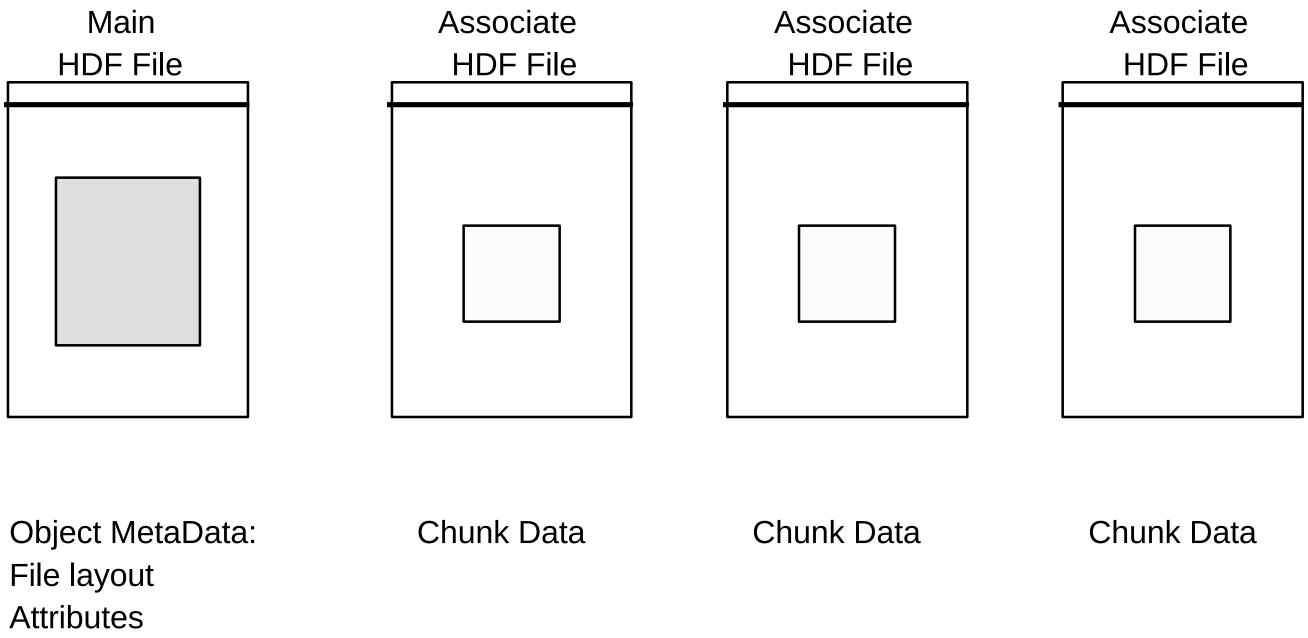
Filesystem

Distirbuted Memory

Networked Processors

Distirbute Filesystems

HDF object distributed over many HDF files



Main
HDF File

Associate
HDF File

Associate
HDF File

Associate
HDF File

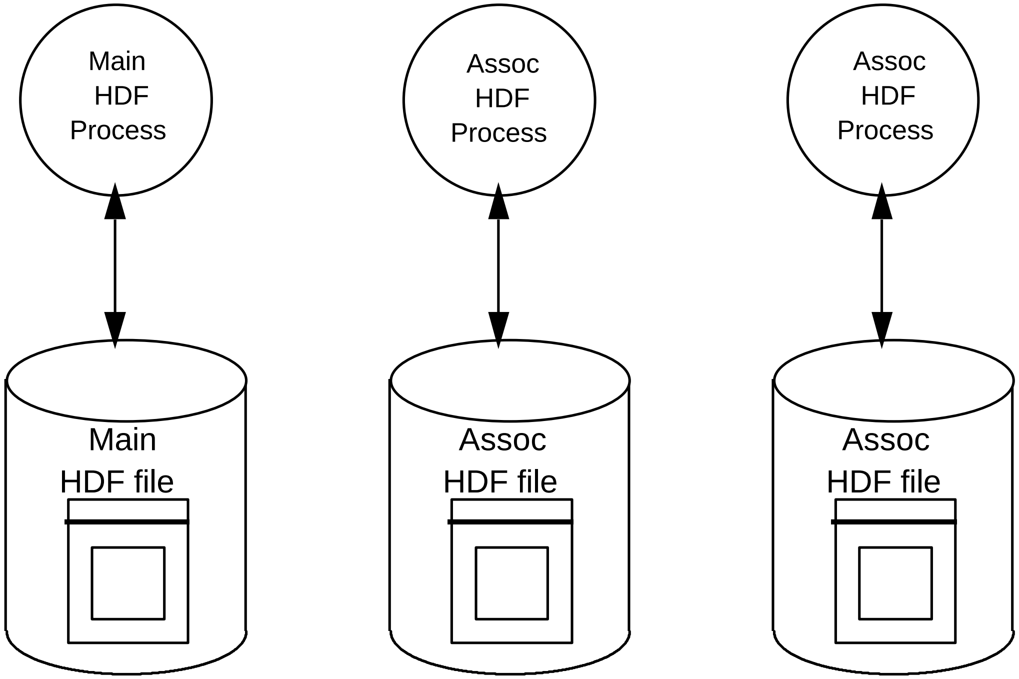
Object MetaData:
File layout
Attributes

Chunk Data

Chunk Data

Chunk Data

Multiple HDF I/O

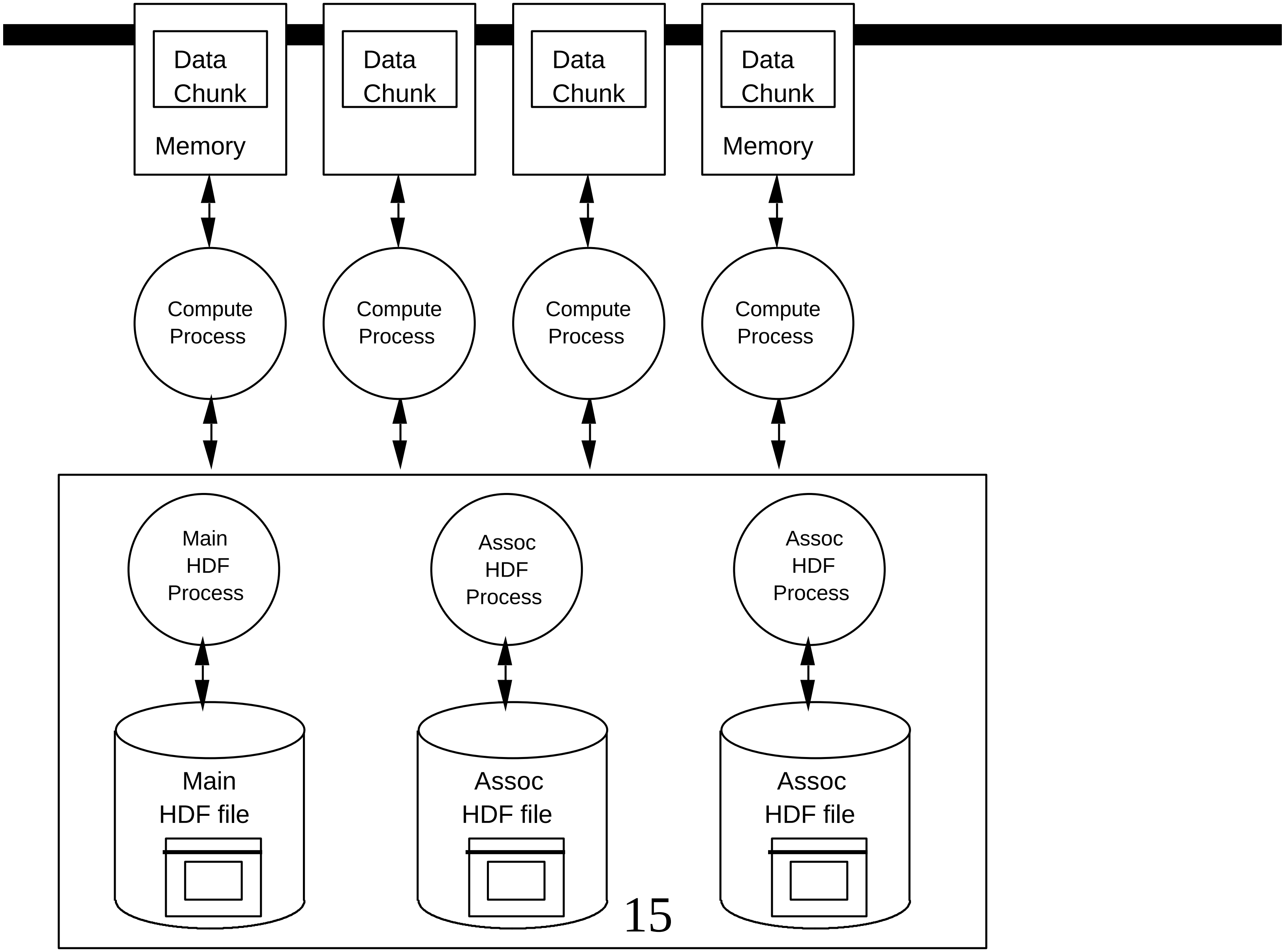


Pros:

Bigger File Capacity
Flexible storage
HDF features
Self Description
Standard Data Format

Cons:
Bigger file sizes

Distributed HDF for a Network of Workstations



Prototype

- **Implementation**
  - **Programming Interface: C & MPI (mpich 1.00.11)**
  - **Workstations networked: Sun, SGI, HP**
- **Test**
  - **A 2 dimension array divided into slices**
  - **Each process holds one slice**
  - **Each process creates an HDF file to store the slice**
  - **Access to various subarrays across processes**

Future Work Plans

- **Variations on current NOW prototype:**
  - **Implement Distributed HDF API**
  - **Performance measurement**
  - **Integrate with compression**
  - **Use HDF external element structure**
- **Next target system is SP2. Homogeneous, but otherwise similar architecture to first prototype.**
- **Distant future: make use of parallel I/O systems (e.g. MPI-IO)**